

REMARKS

This amendment is submitted in response to the Office Action dated January 12, 2006. Reconsideration and allowance of the claims is respectfully requested. In the Office Action, paragraph [0026] is objected to at the first sentence as providing an incomplete sentence. An appropriate edit is supplied herewith.

Claims 1-16 are rejected under 35 USC §101 as directed to non-statutory subject matter. Therefore, the claims have been rewritten as comprising a computer readable medium providing instructions for an obfuscation code. The amended claims do recite statutory subject matter, and the amendments are supported at paragraph [0036], which has been amended to provide an appropriate definition.

Claims 1-20 are rejected under 35 USC §102(b) as being anticipated by "A Taxonomy of Obfuscating Transformations" by Collberg, et al. (hereinafter "Collberg"). Claims 1, 5, 8 and 17-20 are rejected under 35 USC §102(e) as being anticipated by Kiddy (US Patent 6,694,435). These rejections are respectfully traversed.

Collberg clearly fails to disclose the insertion of obfuscating code instructions into a sequence of algorithm instructions where the obfuscating code instructions operate on operands stored on the same registers as operands that are utilized in the execution of the basic algorithm. Obfuscation is achieved by providing, as claimed, an inert executable instruction to first operate on a stored operand, modifying that operand, and then providing further, another inert code instruction elsewhere in the sequence to restore the operand. All of this is done before the code of the basic algorithm is used. The claimed approach makes it extremely difficult for the code to be reversed engineered without extensive and time consuming study. A mere review of the registers operated on by the regular code instructions will leave the reverse engineer to believe that the inert executable code, in fact, provides a significant function. Further, this feature is not taught by Collberg.

Reviewing all of Collberg, and especially the cited section 6.2.1, this reference may be interpreted, at best, as inserting dummy instructions into sections of the sequence of a complete algorithm. However, the teachings are subject to two

restraints. In the first place, Collberg only teaches the insertion of dummy instructions. Collberg does not teach or suggest the use of inert executable instructions, which actually operate on operands that can be found in the registers used by the executing algorithm. Collberg further teaches that it is possible to introduce a bug into a section of the obfuscated code (see the last paragraph of section 6.2.1). However, Collberg then teaches that the code includes a further instruction which ensures that the part of the code with the bug will never be utilized. Therefore, Collberg does not teach or suggest an important claimed feature of the present application, namely, executing instructions outside of the main code to obfuscate the main code.

The claims are further modified to recite that inert executable instructions are redistributed through the executing algorithm. Collberg does not teach or suggest any such redistribution. New claims 21-23 have been added which recite further features of this redistribution and of the execution of both inert executable instructions and algorithm instructions on operands found in the registers by the algorithm. None of these features are taught by Collberg. For these reasons, claims 1 and 17 and all the dependent claims should clearly be allowed over Collberg.

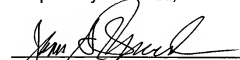
The Examiner also relies on Kiddy to reject several of the claims of the present application. However, Kiddy is subject to the same deficiencies as Collberg. Kiddy teaches only the use of dummy instructions and not the use of inert executable instructions which operate on operands found in registers also used by the main executing code.

The Examiner further relies on Kiddy as teaching reordering of the inert executable instructions within the code, as claimed. However, a further reading of Kiddy's description shows that Kiddy teaches the use of dummy instructions which must be stripped out and dealt with by the compiler. This is clearly disclosed at column 1, lines 50-64 of Kiddy. However, this is the very drawback found in the prior art that is intended to be overcome by the present application. The claimed invention uses inert executable instructions which can be distributed, reordered, and intermixed within the main executing code so that a compiler does not need to detect and strip out the inert instructions prior to execution of the main code. Kiddy clearly teaches non-executing

instructions. Therefore, Kiddy does not teach either one of the significant limitations found in the claims 1 and 17 or the claims dependent thereon.

In view of these clear distinctions, reconsideration and allowance of the claims is respectfully submitted.

Respectfully submitted,



James A. Sheridan

Registration No. 25,435

PATTERSON & SHERIDAN, L.L.P.

3040 Post Oak Blvd. Suite 1500

Houston, TX 77056

Telephone: (713) 623-4844

Facsimile: (713) 623-4846

Attorney for Applicant(s)